НЕЙРОІНФОРМАТИКА ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

NEUROINFORMATICS AND INTELLIGENT SYSTEMS

НЕЙРОИНФОРМАТИКА И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК 004.8:004.032.26

КЛАСТЕРИЗАЦІЯ МАСИВІВ ДАНИХ НА ОСНОВІ КОМБІНОВАНОЇ ОПТИМІЗАЦІЇ ФУНКЦІЙ ЩІЛЬНОСТІ РОЗПОДІЛУ ТА ЕВОЛЮЦІЙНОГО МЕТОДУ КОТЯЧИХ ЗГРАЙ

Бодянський Є. В. – д-р техн. наук, професор, професор кафедри штучного інтелекту, Харківський національний університет радіоелектроніки, Харків, Україна.

Плісс І. П. – канд. техн. наук, провідний науковий співробітник ПНДЛ АСУ, Харківський національний університет радіоелектроніки, Харків, Україна.

Шафроненко А. Ю. – канд. техн. наук, доцент, доцент кафедри інформатики, Харківський національний університет радіоелектроніки, Харків, Україна.

АНОТАЦІЯ

Актуальність. Задача кластеризації масивів спостережень довільної природи є невід'ємною частиною Data Mining, а у більш загальному випадку Data Science, для її вирішення запропонована дуже велика кількість підходів, що відрізняються між собою як апріорними припущеннями що до фізичної природи даних та задачі, так і математичним апаратом. З обчислювальної точки зору задача кластеризації перетворюється у проблему пошуку локальних екстремумів багатоекстремальної функції векторного аргументу щільності за допомогою градієнтних процедур, які багатократно запускаються з різних точок вихідного масиву даних. Пришвидшити процес пошуку цих екстремумів можна, скориставшись ідеями еволюційної оптимізації, що включає в себе алгоритми, інспіровані природою, ройові алгоритми, популяційні алгоритми, тощо.

Мета. Мета роботи полягає у запровадженні процедури кластеризації даних на основі піків щільності розподілу даних та еволюційного методу котячих зграй, що об'єднує в собі основні переваги методів роботи з даними за умов, якщо класи перетинаються, характеризується якісною кластеризацією, високою швидкодією та точністю отриманих результатів.

Метод. Введено метод кластеризації масивів даних на основі комбінованої оптимізації функцій щільності розподілу та еволюційного методу котячих зграй. Перевагою запропонованого підходу є скорочення часу вирішення оптимізаційних задач в умовах коли кластери перетинаються.

Результати. Результати експериментів підтверджують ефективність пропонованого підходу в задачах кластеризації за умов перетинних кластерів та дозволяють рекомендувати запропонований метод для використання на практиці для вирішення проблем автоматичної кластеризації великих даних.

Висновки. Введено метод кластеризації масивів даних на основі комбінованої оптимізації функцій щільності розподілу та еволюційного методу котячих зграй. Перевагою запропонованого підходу є скорочення часу вирішення оптимізаційних задач в умовах коли кластери перетинаються. Метод є досить простим з точки зору чисельної реалізації і не є критичним до вибору оптимізаційної процедури. Результати експериментів підтверджують ефективність запропонованого підходу в задачах кластеризації за умов кластерів, що перетинаються.

КЛЮЧОВІ СЛОВА: нечітка кластеризація, піки щільності розподілу даних, еволюційний метод.

АБРЕВІАТУРИ

SM – режим пошуку (Seeking Mode);

TM – режим трасування (Tracing Mode);

SMP – пошук пулу пам'яті (seeking memory pool);

SRD – крок зміни за кожною координатою простору (seeking range of the selected dimension);

CDC – кількість відстаней, які потрібно змінити (counts of dimension to change).

НОМЕНКЛАТУРА

X – матриця набору даних;

k – номер вектору-спостереження;

i – номер атрибуту вектора-спостереження;

j – номер класу;

$x(\bullet)$ – будь-який вектор-спостереження;

m – кількість неперетинних класів;

DOI 10.15588/1607-3274-2022-4-5

d – відстань між спостереженнями;

σ – параметр ширини – відстань зрізу в прийнятій метриці функції впливу;

$f_G^{x(\bullet)}(x)$ – гаусівська функція;

$f^x(x)$ – функція щільності розподілу даних в масиві;

c – центроїд кластера;

c_p – режим p-го кота;

τ – ітерація пошуку;

β – фаззіфікатор;

α – параметр, який визначає властивості інерції режиму трасування;

η – параметр кроку пошуку;

$\Xi(\tau)$ – випадкова складова, яка вносить додаткові стохастичні рухи у процес трасування;

η_ξ – параметр, який визначає амплітуду рухів

ВСТУП

Задача кластеризації масивів спостережень довільної природи є невід'ємною частиною Data Mining, а у більш загальному випадку Data Science, для її вирішення запропонована дуже велика кількість підходів, що відрізняються між собою як апріорними припущеннями що до фізичної природи даних та задачі, так і математичним апаратом, що використовується [1–4]. З обчислювальної точки зору найбільш простими є, так звані, ієрархічні методи та алгоритми, засновані на розбиттях [3], серед яких слід відзначити процедуру k-середніх, що набула дуже широкого розповсюдження для вирішення найрізноманітніших задач. Тут можна відзначити, що найбільш адекватним математичним апаратом для вирішення задач кластеризації є методи обчислювального інтелекту [5–7] і, перш за все, штучні нейронні мережі, нечіткі системи, еволюційна оптимізація та, так звані, гібридні системи обчислювального інтелекту, що об'єднують ці три напрями.

Об'єкт дослідження кластеризація даних на основі піків щільності розподілу даних та еволюційного методу.

Предмет дослідження процедура оптимізації піків щільності розподілу даних.

Мета роботи полягає у запровадженні процедури кластеризації даних на основі піків щільності розподілу даних та еволюційного методу котячих зграй, що об'єднує в собі основні переваги методів роботи з даними за умов, якщо класи перетинаються, характеризується якісною кластеризацією, високою швидкодією та точністю отриманих результатів.

1 ПОСТАНОВКА ЗАВДАННЯ

Вихідною інформацією для вирішення задачі кластеризації традиційно є матриця спостережень $X = \{x(1), x(2), ..., x(k), ..., x(N)\}$, $x(k) = \{x_i(k)\} \in R^n$, при цьому дані попередньо відцентровано на гіперкуб так, що $x(k) = \{x_{i,i_2}(k)\} \in R^{n_1 \times n_2}$. Така ситуація може виникати у випадку обробки масивів зображень.

2 ОГЛЯД ЛІТЕРАТУРИ

Тут слід відзначити, що в загальному випадку вирішення задачі кластеризації суттєво ускладняється, якщо вихідні вектори (тут у загальному випадку матриці) спостереження мають велику різноманітність, викривлені збуреннями та завадами, містять пропуски, самі вихідні масиви або занадто великі (Big Data) або занадто короткі, кластери можуть мати досить складну форму, а їх кількість апріорі невідома.

У цьому випадку найбільш ефективними (але й найбільш складними) є алгоритми, що базуються на аналізі щільностей розподілу даних, серед яких в якості одного найбільш «популярних» є DENCLUE [9] та його модифікації [10–12], що були запропоновані для вирішення задач кластеризації великих масивів векторних даних високої розмірності, при цьому класи, що формуються у процесі кластеризації, можуть мати будь яку складну форму. В основі цих алгоритмів полягає пошук екстремумів максимумів функції щільності розподілу даних у масиві, що аналізується (багатоекстремальна оптимізація), при цьому ця функція формується, як суперпозиція ядерних (дзвонуватих) функцій, пов'язаних з кожним спостереженням. Фактично ця функція будується на основі вікон Парзена [13] та оцінок Надарая-Ватсона [14, 15].

З обчислювальної точки зору задача кластеризації перетворюється у проблему пошуку локальних екстремумів багатоекстремальної функції векторного аргументу щільності за допомогою градієнтних процедур, які багатократно запускаються з різних точок вихідного масиву даних. Зрозуміло, що це займає досить багато часу, оскільки апріорі навіть невідомо скільки ж екстремумів має сформована функція щільності.

Пришвидшити процес пошуку цих екстремумів можна, скориставшись ідеями еволюційної оптимізації, що включає в себе алгоритми, інспіровані природою, ройові алгоритми, популяційні алгоритми, тощо [16–18]. При цьому пошук ведеться одночасно групою агентів, що діють або незалежно, або у взаємодії, що дозволяє суттєво пришвидшити процес пошуку екстремумів, кожен з яких «відповідає» тому або іншому кластеру, що формується.

3 МАТЕРІАЛИ І МЕТОДИ

Основними поняттями, на яких базується DENCLUE є функція впливу, функція щільності та атрактори щільності, що за суттю є локальними екстремумами функції щільності.

У загальному випадку функція впливу для будь-якого векторного спостереження $x(\bullet)$ з вихідного масиву X є ядерною дзвонуватою функцією $f^{x(\bullet)}(x)$, при цьому найбільш популярною є традиційна гаусівська функція

DOI 10.15588/1607-3274-2022-4-5

$$f_G^{x(\bullet)}(x)=\exp\left(-\frac{d^2(x,x(\bullet))}{2\sigma^2}\right)=\exp\left(-\frac{\|x-x(\bullet)\|^2}{2\sigma^2}\right) \quad (1)$$

(тут $d^2(x,x(\bullet))$ – евклідова відстань, σ^2 – параметр ширини функції впливу), завдяки простоті обчислення її похідних.

У матричному випадку замість евклідової можна використати метрику Флобеніуса, при цьому функція впливу набуває вигляду

$$\begin{aligned} f_G^{x(\bullet)}(x)&=\exp\left(-\frac{d^2(x,x(\bullet))}{2\sigma^2}\right)= \\ &=\exp\left(-\frac{Tr\left(x-x(\bullet)\right)\left(x-x(\bullet)\right)^T}{2\sigma^2}\right), \end{aligned} \quad (2)$$

де $Tr(\bullet)$ – символ сліду матриці.

Нескладно бачити, що (2) є узагальненням (1).

На основі функцій впливу формується функція щільності розподілу даних у масиві X у вигляді

$$f^x(x)=\sum_{k=1}^{N} f\left(x,x(k)\right), \quad (3)$$

що по сутті є оцінкою Надарая-Ватсона. Нескладно бачити, що функція $f^x(x)$ може приймати значення в інтервалі

$$1\le f^x(x)\le N,$$

при цьому крайні значення з цього інтервалу приймаються, коли вибірка містить лише одне спостереження або усі N спостережень співпадають, тобто існує лише один кластер – вироджена ситуація.

Для знаходження $m>1$ кластерів необхідно ввести у розгляд деякий поріг $\xi>1$, що дозволяє формувати дійсно значущі кластери, відстежуючи аномальні спостереження та класи, що містять занадто мало даних.

Власне процес формування кластерів пов'язаний з відшуканням усіх екстремумів функції щільності (3) за допомогою градієнтної процедури

$$\begin{aligned} x^l&=x^{l-1}+\eta^l\frac{\nabla f^x\left(x^l,x^{l-1}\right)}{\left\|\nabla f^x\left(x^l,x^{l-1}\right)\right\|}, \\ x_0&=x(k), l=0,1,2,...;\forall k=1,2,...,N, \end{aligned} \quad (4)$$

тобто кількість запусків алгоритму (4) визначається обсягом навчальної вибірки N. Зрозуміло, що при великих N процес кластеризації – пошуку локальних екстремумів може потребувати дуже багато часу. Тому запропоновані модифікації DENCLUE пов'язані з пришвидшенням процесу пошуку локальних екстремумів (3) шляхом модифікації градієнтної процедури (4) [10–12].

У випадку коли спостереження $x(k)$ у вибірці $X\in(n_1\times n_2)$-матриці, нескладно ввести у розгляд матричний варіант процедури (4):

$$x^l=x^{l-1}+\eta^l\Gamma^x\left(x,x^{l-1}\right)\left(Tr\Gamma^x\left(x,x^{l-1}\right)\Gamma^{xT}\left(x,x^{l-1}\right)\right)^{-\frac{1}{2}},$$

де $\Gamma^x\left(x,x^{l-1}\right)=\left\{\frac{\partial f^x\left(x,x^{l-1}\right)}{\partial x_{i_1i_2}}\right\}\in R^{n_1\times n_2}$.

Процес градієнтної оптимізації закінчується відшуканням m локальних екстремумів функції (3), при цьому чим менше значення ξ, тим більше кластерів може бути сформовано.

Пришвидшити процес відшукання локальних екстремумів можна, використовуючи замість градієнтного пошуку методи еволюційної оптимізації, серед яких в якості достатньо ефективного, чисельно простого і швидкого можна відзначити, так званий, пошук на основі котячих зграй, що повинен бути модифікований для вирішення задачі кластеризації.

Для пошуку глобального екстремуму скалярної функції $f(x)$ векторного аргументу $x=(x_1,x_2,...,x_n)^T\in R^n$ авторами [18, 19] було запропоновано використовувати модель поведінки котячих зграй (cat swarm – CS) при цьому передбачається, що кожен кіт cat_p зграї, яка складається з Q особин ($p=1,2,...,Q$), може бути в одному з двох станів: режиму пошуку (Seeking Mode – SM) и режимі погоні (Tracing Mode – TM). При цьому режим пошуку пов'язаний з повільними рухами з незначною амплітудою біля вихідної позиції (сканування простору в поточній позиції), а режим погоні визначається швидкими стрибками з великою амплітудою і дозволяє вивести кота cat_p з локального екстремуму, якщо він туди потрапив. Поєднання локального сканування та різких змін поточного стану дозволяє з більшою ймовірністю відшукати глобальний екстремум у порівнянні з традиційними методами багатоекстремальної оптимізації.

Процес відшукання екстремуму за допомогою котячої зграї може бути реалізований у вигляді наступної послідовності кроків:

Крок CS 1: створити зграю з Q котів у вигляді набору n-вимірних векторів $x_p^{(0)}$, випадковим чином розподілених на безліч допустимих значень аргументів R_x^n, тобто $x_p^{(0)}\in R_x^n\subset R^n$; оцінити значення оптимізованої функції (фітнес-функції) $f(x_p(0))$ у всіх Q точках, при цьому передбачається, що метою оптимізації є відшукання глобального мінімуму $f(x)$.

DOI 10.15588/1607-3274-2022-4-5

Крок CS 2: ввести параметр стану SPC (self position consideration), який приймає два значення 1 або 0; випадково розділити зграю на дві групи: коти в пошуку (SPC=1) і коти в погоні (SPC=0).

Крок CS 3: якщо SPC=1, запустити відповідну групу котів у пошуку, які залишилися коти с SPC=0 запустити в режим погоні.

Крок CS 4: оцінити значення фітнес-функції та зберегти нові стани $x_p(1)$, відповідні найменшим значенням $f(x_p(1))$.

Крок CS 5: повернутися до кроку CS 1 з оновленою зграєю $x_p(1), p = 1,2,...,Q.$

Режими пошуку та погоні можуть бути реалізовані паралельно і також складатися з послідовності кроків. При цьому режим пошуку котячої зграї відповідає процесу локального пошуку завдання оптимізації. Режим пошуку визначається трьома основними факторами: обсягом пам'яті пошуку (seeking memory pool – SMP), який визначає кількість створюваних копій кожного кота cat_p, кроком зміни за кожною координатою простору R_x^n (seeking range of the selected dimension – SRD) та змінюваних координат (counts of dimension to change – CDC). Власне, режим пошуку може бути реалізований у вигляді наступної послідовності кроків:

Крок SM 1: якщо SPC = 1, створити C (C=SMP) копій cat_p.

Крок SM 2: відповідно до прийнятого CDC змінити стан cat_p.

Крок SM 3: оцінити значення оптимізованої фітнес-функції для кожного зміненого стану cat_p.

Крок SM 4: ввести ймовірність вибору кожного зміненого стану

$$P_p = \frac{f(x_p(\tau)) - f_{\min}(x_p(\tau))}{f_{\max}(x_p(\tau)) - f_{\min}(x_p(\tau))}, \tau = 1,2,...,T \qquad (5)$$

та кота з максимальним значенням P_p виключити з подальшого розгляду. Кіт з $P_p = 0$ є «найкращою» копією cat_p, оскільки їй відповідає найменше значення оптимізованої функції $f_{\min}(x_p(\tau))$.

Режим погоні відповідає процесу глобального пошуку, що дозволяє «проскакувати» локальні екстремуми оптимізованої функції, і може бути реалізований у вигляді послідовності кроків:

Крок TM 1: якщо SPC = 0, для групи котів в погоні розрахувати для кожного cat_p швидкості руху за кожною координатою за допомогою рекурентного виразу

$$\upsilon_{pi}(\tau+1) = \upsilon_{pi}(\tau) + r(\tau)\eta_{TM}(x_{best,i}(\tau) - x_{pi}(\tau)), \qquad (6)$$

де $\upsilon_{pi}(\tau)$ – швидкість руху p-го кота по i-й координаті на τ-й ітерації погоні, $0 < r(\tau) < 1$ – випадковий параметр погоні, η_{TM} – постійний крок погоні, $x_{best,i}(\tau)$ – найкраще вирішення задачі оптимізації, отримане на τ-й ітерації.

Крок TM 2: ввести гранично можливі значення швидкостей $\upsilon_{\min}$ і $\upsilon_{\max}$, для кожного кота перевірити умову

$$\upsilon_{\min} < \upsilon_{pi}(\tau+1) < \upsilon_{\max}$$

і якщо воно порушується, покласти $\upsilon_{pi}(\tau+1)$ рівним відповідному значенню $\upsilon_{\min}$ і $\upsilon_{\max}$.

Крок TM 3: змінити положення кожного кота в погоні відповідно до співвідношення

$$x_{pi}(\tau+1) = x_{pi}(\tau) + \upsilon_{pi}(\tau). \qquad (7)$$

Крок TM 4: перевірити, чи належить $x_p(\tau+1)$ R_{xi}^n.

Можна помітити, що розглянутий алгоритм пошуку реалізує по суті покоординатний спуск (метод Гаусса-Зейделя), що вимагає багаторазового оцінювання оптимізованих значень і характеризується низькою швидкістю збіжності. У режимі погоні реалізується градієнтний пошук з великим кроком, що в загальному випадку не гарантує пошуку глобального екстремуму. У зв'язку з цим доцільно модернізувати процедуру оптимізації на основі котячих зграй шляхом її рандомізації на основі випадкового пошуку [21–22], що володіє цілою низкою переваг перед детермінованими процедурами пошуку екстремуму.

Оскільки режим пошуку SM є по суті процесом локальної оптимізації, рух кожного з котів cat_p с SPC=1 доцільно організувати в антиградієнтному напрямку відповідно до стандартної рекурентної градієнтної процедури

$$x_p(\tau+1) = x_p(\tau) - \eta_{SM}\hat{\nabla}f(x_p(\tau)), \qquad (8)$$

де $\hat{\nabla}f(x_p(\tau))$ – оцінки градієнта оптимізованої функції у точці $x_p(\tau)$, η_{SM} – крок пошуку у просторі R_x^n.

Складові градієнта $\nabla f(x_p(\tau))$, є частковими похідними $\frac{\partial f(x_p(\tau))}{\partial x_p}$, можуть бути оцінені шляхом вимірювання оптимізованої функції в пробних станах в околиці точки $x_p(\tau)$. Найбільш простим з обчислювальної точки зору є пошук з центрального пробою [19], при цьому проводиться оцінка оптимізованої функції $(n+1)$-й точці $(CDC = n): x_p(\tau)$, $x_p(\tau) + \eta_{SRD}e_1$,

DOI 10.15588/1607-3274-2022-4-5

$x_p(\tau)+\eta_{SRD}e_2,...,x_p(\tau)+\eta_{SRD}e_n$, де e_i – координатні орти, η_{SRD} – розмір пробного кроку, який визначається прийнятим значенням SRD.

Встановивши $n+1$ значення функції $f(x_p(\tau))$, $f(x_p(\tau)+\eta_{SRD}e_2),...,f(x_p(\tau)+\eta_{SRD}e_n)$, замість градієнта $\nabla f(x_p(\tau))=\left(\frac{\partial f(x_p(\tau))}{\partial x_{p1}},\frac{\partial f(x_p(\tau))}{\partial x_{p2}},...,\frac{\partial f(x_p(\tau))}{\partial x_{pn}}\right)^T$, можна ввести його оцінку $\hat{\nabla} f(x_p(\tau))$ з компонентами

$$\frac{\partial \hat{f}(x_p(\tau))}{\partial x_{pi}}=\frac{1}{\eta_{SRD}}\left(f(x_p(\tau))+\eta_{SRD}e_i\right)-f(x_p(\tau)),i=1,2,...,n.$$

Реалізувавши далі крок у просторі R_x^n відповідно до (8), приходимо до нового стану cat_p в режимі пошуку з координатами

$$\begin{cases} x_{p1}(\tau+1)=x_{p1}(\tau)-\frac{\eta_{SM}}{\eta_{SRD}}(f(x_p(\tau))+\eta_{SRD}e_1)-f(x_p(\tau)),\\ x_{p2}(\tau+1)=x_{p2}(\tau)-\frac{\eta_{SM}}{\eta_{SRD}}(f(x_p(\tau))+\eta_{SRD}e_2)-f(x_p(\tau)),\\ x_{pn}(\tau+1)=x_{pn}(\tau)-\frac{\eta_{SM}}{\eta_{SRD}}(f(x_p(\tau))+\eta_{SRD}e_n)-f(x_p(\tau)). \end{cases} \quad (9)$$

Як недолік цієї процедури оптимізації можна відзначити фіксоване значення $CDC=n$, що вимагає послідовної зміни всіх координат cat_p в просторі R_x^n. Розширити можливості процесу пошуку можна, звернувшись до рандомізованих процедур [20–21], найпростішою з яких є суто випадкова оцінка спускового напрямку, сенс якого полягає в тому, що зі стану $x_p(\tau)$ робиться випадкова проба $x_p(\tau)+\eta_{SRD}\Xi$, де $\Xi=(\xi_1,\xi_2,...,\xi_n)^T$ – одиничний випадковий вектор, рівномірно розподілений у просторі R_x^n. У разі якщо $x_p(\tau)+\eta_{SRD}\Xi<f(x_p(\tau))$, робиться робочий крок пошуку

$$x_p(\tau+1)=x_p(\tau)-\eta_{SM}\Xi \quad (10)$$

(при цьому можна прийняти $\eta_{SRD}=\eta_{SM}$), в іншому випадку проба визнається невдалою та реалізується спроба з новим вектором Ξ.

Узагальненням цієї процедури є оцінка напряму пошуку за найкращою з кількох випадкових спроб. При цьому з вихідного стану $x_p(\tau)$ робиться кілька випадкових проб оптимізованої функції $x_p(\tau)+\eta_{SRD}\Xi_l$ у випадкових напрямках $\Xi_l(l=1,2,...,n,...,L)$, при цьому фактор CDC може перевищувати значення n. За напрямок спуску вибирається той напрямок, що забезпечило найменше значення функції $f(x_p)$, тобто cat_p переводиться в новий стан згідно з виразом

$$x_p(\tau+1)=x_p(\tau)+\eta_{SRD}\Xi^*. \quad (11)$$

Зауважимо також, що при $L=1$, процедури (10) і (11) співпадають.

Об'єднуючи процедури пошуку (8), (9), (11), можна ввести до розгляду пошук на основі статичного градієнта. У цьому випадку за оцінку градієнта приймається середньозважене L випадкових напрямків, кожен з яких береться з вагою, що відповідає варіації $f(x_p)$ вздовж цього напряму:

$$\hat{\nabla} f(x_p(\tau))=-\frac{\sum_{l=1}^{L}\Xi_l\left(f\left(x_p(\tau)+\eta_{SRD}\Xi_l\right)-\nabla f(x_p(\tau))\right)}{\left\|\sum_{l=1}^{L}\Xi_l\left(f\left(x_p(\tau)+\eta_{SRD}\Xi_l\right)-\nabla f(x_p(\tau))\right)\right\|}. \quad (12)$$

Підставляючи далі (12) (11), отримуємо процедуру градієнтного спуску в напрямку мінімуму функції, що оптимізується. Таким чином, всі коти з SPC=1 зміщуються в напрямку локальних мінімумів функції, що оптимізується.

Режим погоні ТМ на відміну від локального режиму пошуку SM забезпечує загальну процедуру оптимізації на основі CS глобальні властивості якої дозволяють не застрягати їй у локальних екстремумах. Зрозуміло, що крім процедури (5), (6) існують інші алгоритми, що володіють необхідними властивостями.

Одним із таких найбільш ефективних чисельно простих алгоритмів є метод важкої кульки, що спирається на аналогію руху важкого тіла по викривленій поверхні з урахуванням сил тяжіння та тертя. При цьому через інерцію кулька-кіт «проскакує» локальні екстремуми, а через тертя рух має зупинитися в глобальному екстремумі.

Даний алгоритм для котів у режимі погоні (SPC=0) може бути записаний у вигляді

$$\begin{aligned} x_p(\tau+1)=x_p(\tau)-\alpha(x_p(\tau)-\\ -x_p(\tau-1))-\eta_{TM}\hat{\nabla} f(x_p(\tau)), \end{aligned} \quad (13)$$

де α – параметр, що визначає інерційні властивості процесу гонитви. При $\alpha=0$ (13) повністю збігається з (8), відрізняючись лише кроком η_{SM}. При $\alpha=1$ процес погоні стає незагасаючим, тому цей параметр вибирається в інтервалі $0<\alpha<1$, при цьому чим ближче α до одиниці, тим сильніше виявляються інерційні властивості, проте процес слабо згасає в околиці екстремуму. У зв'язку з цим доцільно кожному коту з SPC=0 призначити різні значення параметра α.

Зауважимо також, що в процедуру (13) може бути введена випадкова компонента, що вводить додаткове

DOI 10.15588/1607-3274-2022-4-5

«рискання» в процес гонитви, що покращує глобальні властивості алгоритму. При цьому (13) модифікується до вигляду

$$x_p(\tau+1) = x_p(\tau) - \alpha(x_p(\tau) - x_p(\tau-1)) - \\ -\eta_{TM}\hat{\nabla}f(x_p(\tau)) + \eta_{SRD}\Xi,$$

тобто cat_p одночасно знаходиться і в режимі погоні, і в режимі пошуку-сканування простору R_x^n.

4 ЕКСПЕРИМЕНТИ

Дослідження методу кластеризації масивів даних на основі комбінованої оптимізації функцій щільності розподілу та еволюційного методу котячих зграй проводились на трьох навчальних вибірках: Spambase, Page Blocks, Iris та Ecoli.

Таблиця 1 – Характеристики наборів даних

Вибірка	Кількість спостережень	Кількість атрибутів
Spambase	4601	57
Page Blocks	5472	10
Iris	150	4
Wine	178	13
Ecoli	336	8

Якість роботи метода кластеризації даних на основі піків щільності розподілу даних та еволюційного методу котячих зграй (Proposed method – PM) перевірялась за допомогою основних оцінок якості кластеризації. Існує кілька метрик для оцінки якості кластеризації. Всі метрики, що використовуються для оцінки запропонованого методу базуються на так званому методі оцінювання за допомогою золотого стандарту(golden set).

1. Метрика чистоти кластеризації (purity – Pur). Для обчислення даного показника кожному кластеру присвоюється клас, з яким у кластера максимальне перекриття по привласненим об'єктам. Після присвоєння міток класів обчислюється правильність даної кластеризації як число об'єктів класу, з яким асоційований кластер, поділене на загальне число об'єктів в кластері. У цьому сенсі дана метрика схожа на показник точності класифікації.

2. Метрика нормованої взаємної інформації (normalized mutual information – NMI). Дана метрика заснована на понятті ентропії.

3. Коефіцієнт Ренда (rand index – RI). Даний підхід до оцінки якості кластеризації перегукується з методами оцінки якості алгоритмів класифікації.

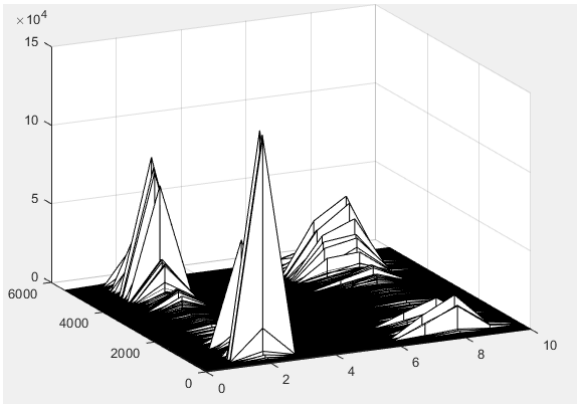

Рисунок 1 – Навчальна вибірка Page Blocks

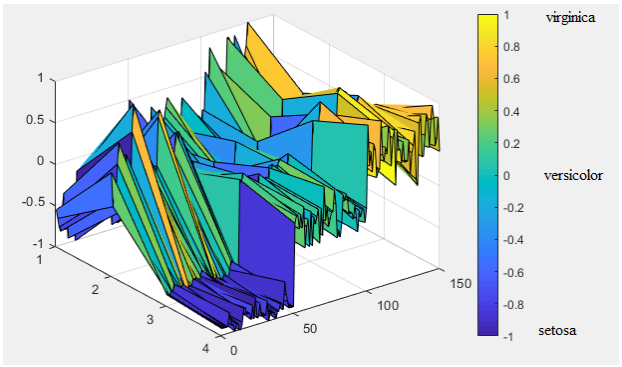

Рисунок 2 – Навчальна вибірка Iris

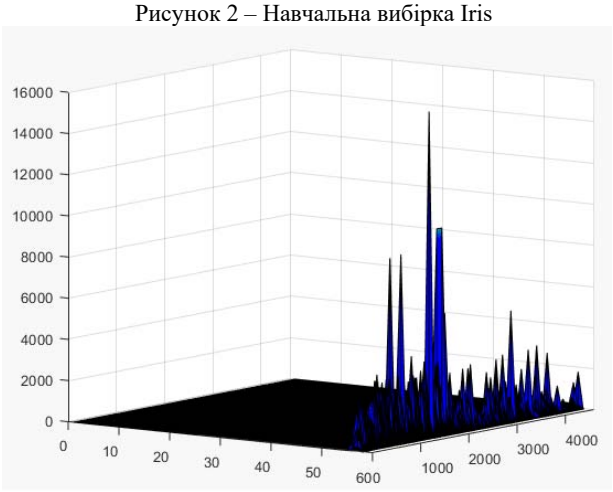

Рисунок 3 – Навчальна вибірка Spambase

Порівняльний аналіз проводився з відомими методами кластеризації даних, такими як FCM, DBSCAN та CLARA.

5 РЕЗУЛЬТАТИ

Результати кластерного аналізу даних на вибірках Page block, Wine, Iris, Ecoli та Spambase, за показниками оцінки якості кластеризації наведено на рисунках нижче.

На гістограмах продемонстровані результати кластерного аналізу за якими можна зробити висновок, що запропонований метод кластеризації даних на основі піків щільності розподілу даних та еволюційного методу котячих зграй дає оцінку кластеризації вище, ніж більш відомі методи кластеризації завдяки оптимізаційній процедурі еволюційного алгоритму.

DOI 10.15588/1607-3274-2022-4-5

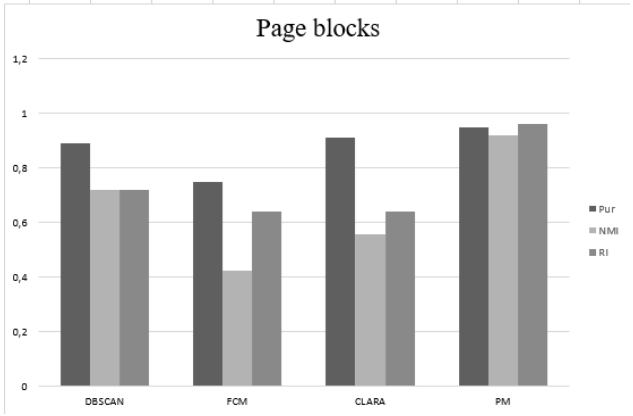

Рисунок 4 – Показники якості кластеризації вибірки Page Blocks

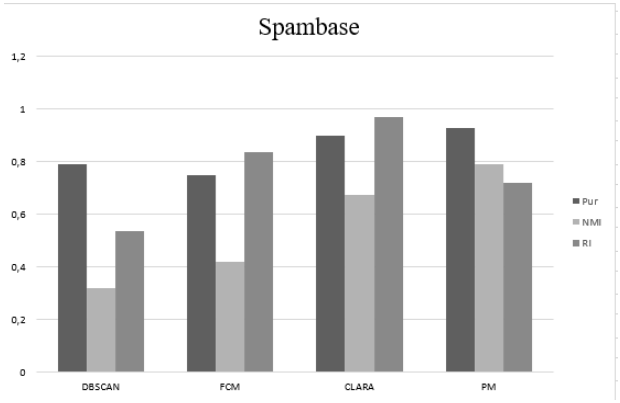

Рисунок 5 – Показники якості кластеризації вибірки Spambase

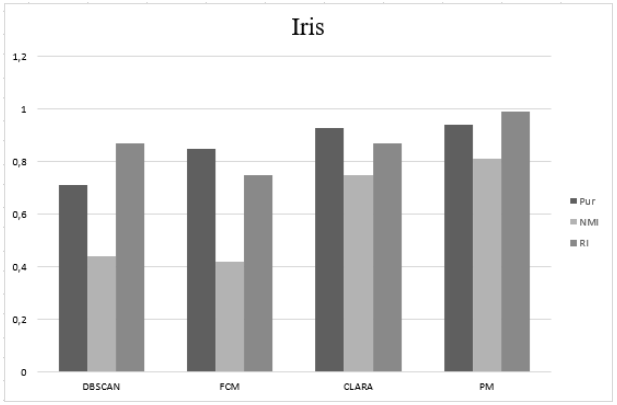

Рисунок 6 – Показники якості кластеризації вибірки Iris

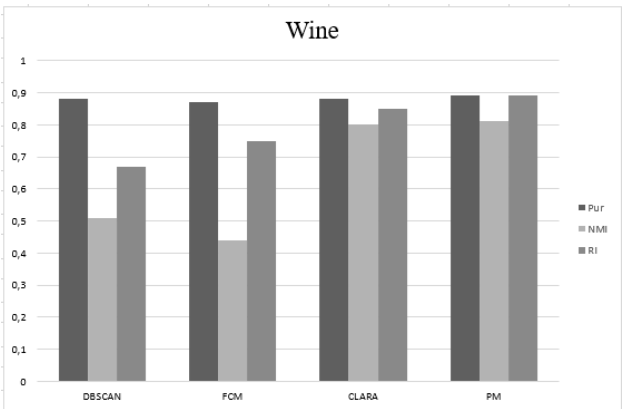

Рисунок 7 – Показники якості кластеризації вибірки Wine

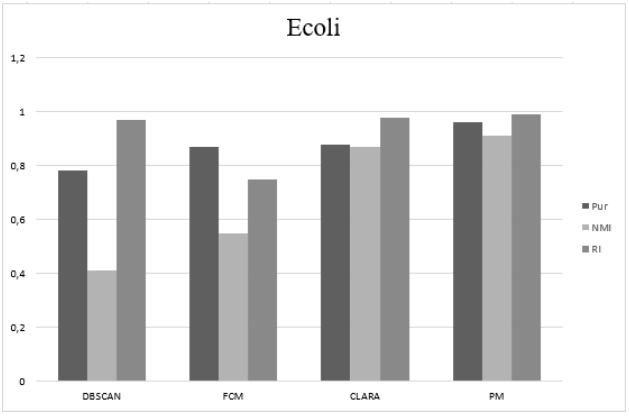

Рисунок 8 – Показники якості кластеризації вибірки Ecoli

Крім аналізу якості кластеризації даних, потрібно оцінити швидкість роботи методу. Якість методу кластеризації повинна відповідати швидкості і простоти з точки зору математичних розрахунків.

Проведено аналіз методу кластеризації даних на основі піків щільності розподілу даних та еволюційного методу котячих зграй на 100 спостереженнях різних вибірок даних.

На рисунках, що представлені нижче наведений порівняльний результат швидкості роботи методів кластеризації.

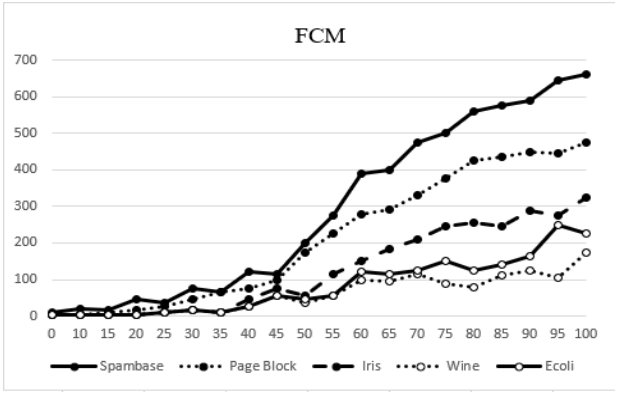

Рисунок 4 – швидкість роботи FCM (в мсек.)

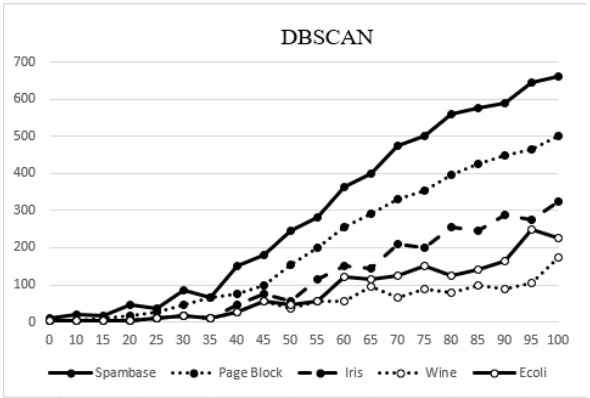

Рисунок 5 – швидкість роботи DBSCAN (в мсек.)

DOI 10.15588/1607-3274-2022-4-5

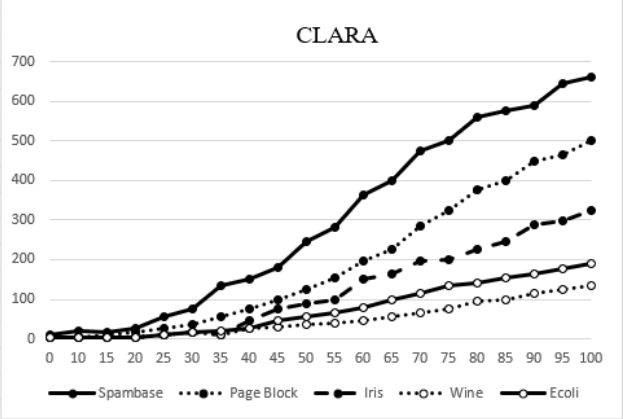

Рисунок 6 – Швидкість роботи CLARA (в мсек.)

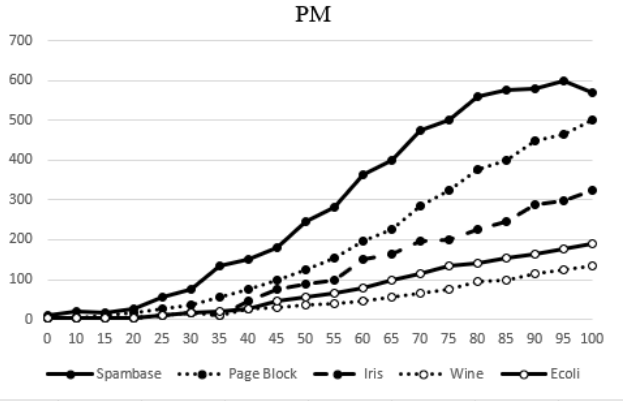

Рисунок 7 – Швидкість роботи PM (в мсек.)

6 ОБГОВОРЕННЯ

Аналізуючи результати отриманих експериментальних досліджень, що проводились на п’яти різної природи даних із вибірок UCI репозиторію та порівняльного аналізу роботи методу кластеризації даних на основі піків щільності розподілу даних та еволюційного методу котячих зграй із методами кластеризації, що базуються як на класичному підході до кластеризації даних, так і більш екзотичних: DBSCAN та CLARA, що використовують аналіз щільності даних які підлягають кластеризації, запропонований метод демонструє достатньо високі результати.

Основними перевагами запропонованого методу полягає в простоті математичних розрахунків, швидкості роботи з даними, незалежно від виду, розміру та якості вибірки, що аналізується. Порівняльний результат швидкості роботи методів кластеризації експериментальних досліджень наведений на графіках, які наочно демонструють швидкість роботи методів на різних вибірках. Слід відзначити точність роботи метода кластеризації даних на основі піків щільності розподілу даних та еволюційного методу котячих зграй та отриманих результатів кластеризації, що досягається за допомогою оптимізаційної процедури еволюційного алгоритму. Як видно із рисунків 4–8, показники якості кластеризації на різних вибірках запропонованого методу достатньо високий, незалежно від метрики, що використовуються для оцінки методів кластеризації.

DOI 10.15588/1607-3274-2022-4-5

ВИСНОВКИ

Введено метод кластеризації масивів даних на основі комбінованої оптимізації функцій щільності розподілу та еволюційного методу котячих зграй. Перевагою запропонованого підходу є скорочення часу вирішення оптимізаційних задач в умовах коли кластери перетинаються. Метод є досить простим з точки зору чисельної реалізації і не є критичним до вибору оптимізаційної процедури. Результати експериментів підтверджують ефективність запропонованого підходу в задачах кластеризації за умов перетинних кластерів.

Наукова новизна: вперше запропонований кластеризації масивів даних на основі комбінованої оптимізації функцій щільності розподілу та еволюційного методу котячих зграй

Практичне значення: результати експерименту дозволяють рекомендувати запропонований метод для використання на практиці для вирішення проблем автоматичної кластеризації багатоекстремальних даних з різною щільністю в умовах класів, що перетинаються.

Перспективи подальших досліджень методи нечіткої кластеризації даних для широкого класу практичних проблем.

ПОДЯКИ

Робота виконана в рамках науково-дослідного проєкту державного бюджету Харківського національного університету радіоелектроніки «Глибокі гібридні системи обчислювального інтелекту для аналізу потоків даних та їх швидке навчання» (номер державної реєстрації 0119U001403).

ЛІТЕРАТУРА/ЛИТЕРАТУРА

1. Gan G. Data Clustering: Theory, Algorithms and Applications / G. Gan, Ch. Ma, J. Wu. – Philadiphia, Pensilvania : SIAM, 2007. – 455 p. DOI: https://doi.org/10.1137/ 1.9780898718348
2. Abonyi J. Cluster Analisis for Data Mining and System Identification / J. Abonyi, D. Feil. – Basel : Birlhause, 2007. – 303 p.
3. Xu R. Clustering/ R. Xu, D. C. Wunsch. – Hoboken N.J. : John Wiley & Sons, Inc., 2009. – 398 p.
4. Aggarwal C. C. Data Mining / C. C. Aggarwal. – Switzerland : Springer, 2015. – 727 p. DOI https://doi.org/ 10.1007 / 978-3-319-14142-8.
5. Engelbrecht A. Computational intelligence: an introduction / A. Engelbrecht. – Sidney : John Wiley & Sons, 2007. – 597 p.
6. Rutkowski L. Computational Intelligence Methods and Techniques / L Rutkowski. – Berlin Heidelberg : Springer-Verlag, 2008. – 514 p.
7. Kroll A. Computational Intelligence. Eine Einfürung in Problelme, Methoden and Tchnische Anwendungen / A. Kroll. – München : Oldenbourg Verlag, 2013. – 428 p.
8. Kohonen T. Self-Organizing Maps/ T. Kohonen. – Berlin : Springer, 1995. – 362 p. DOI: 10.1007/978-3-642-56927-2.
9. Hinneburg A. An efficient approach to clustering in large multimedia databases with noise / A. Hinneburg, D. Klein // Proc. 4th Int. Conf. in Knowledge Discovering and Data Mining – KDD98, N.Y. : AAAI Press, Aug. 27, 1998. – Hinneburg, 1998. – P. 58–65.
10. Hinneburg A. Denclue 2.0: Fast clustering based on kernel density estimation/ A. Hinneburg, H. H. Gabriel // International symposium on intelligent data analysis. – Springer, Berlin, Heidelberg, 2007. – P. 70–80. https://doi.org/10.1007 /978-3-540-74825-0_7

11. Hinneburg A. A general approach to clustering in large databases with noise-knowledge and Identification Systems / A. Hinneburg, D. A. Keim – 2003. – 5 (4). – P. 387–415. https://doi.org/10.1007 /s10115-003-0086-9
12. Rehioui H. DENCLUE-IM: A new approach for big data clustering/ H. Rehioui et al. // Procedia Computer Science. – 2016. – Vol. 83. – P. 560–567. DOI: 10.1016/j.procs.2016.04.2 65
13. Epanechnikov V. A. Nonparametric estimation of multivariate probability density / V. A. Epanechnikov // Probability theory and its Application – 1968 – 14 – №2 – P. 156–161.
14. Nadaraya E. A. On non-parametric estimates of density functions and regression curves/ E. A. Nadaraya // Theory of Probability & Its Applications. – 1965. – Т. 10, №. 1. – P. 186–190.
15. Watson G. S. Smooth regression analysis/ G. S Watson // Sankhyā: The Indian Journal of Statistics, Series A. – 1964. – P. 359–372.
16. Grosan C. Swarm intelligence in Data Mining / C. Grosan, A. Abraham, M. Chis // Studies in Computational Intelligence. – 2006. – №34. – P. 1–20.
17. The Fast Modification of Evolutionary Bioinspired Cat Swarm Optimization Method [Electronic resource]/ [A. Yu. Shafronenko, Ye. V. Bodyanskiy, I. P. Pliss] // 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL),2019. – Sozopol, Bulgaria, 2019. – P. 548–552. DOI: 10.1109 /CAOL46282. 2019.9019583
18. Eiben A. Introduction to Evolutionary Computing / A. Eiben, J. Smith – Heidelberg : Springer, 2003.
19. Karpenko A. P. Population algorithms for global continious optimization. Review of new and little-known algorithms / A. P. Karpenko // Supplement to the journal “Information Technologies”. – №7/2012. – 32 p.
20. Chu S.-C. Cat swarm optimization / S.-C. Chu, P.-W. Tsai, J. S. Pan // Lecture Notes in Artificial Intelligence. – 4099. – Berlin Heidelberg : Springer-Verlag, 2006. – P. 854–858.
21. Chu S.-C. Computational Intelligence based on the behavior of cats / S.-C. Chu, P.- W. Tsai // International Journal of Innovative Computing, Information, and Control. – 2007. –Vol. 3, №1. – P.163–173.
22. Shafronenko A. Online Credibilistic Fuzzy Clustering Method Based on Cauchy Density Distribution Function / [A. Shafronenko, Ye. Bodyanskiy, I. Pliss, I. Klymova] // 2021 11th International Conference on Advanced Computer Information Technologies (ACIT) : proceedings. – Deggendorf, Germany : IEEE, 2021. – P.704–707. DOI: 10.1109/ ACIT52158.2021.9548572

Received 03.09.2022.
Accepted 02.11.2022.

УДК 004.8:004.032.26

КЛАСТЕРИЗАЦИЯ МАССИВОВ ДАННЫХ НА ОСНОВЕ КОМБИНИРОВАННОЙ ОПТИМИЗАЦИИ ФУНКЦИЙ ПЛОТНОСТИ РАСПРЕДЕЛЕНИЯ И ЭВОЛЮЦИОННОГО МЕТОДА КОШАЧЬИХ СТАЙ

Бодянский Е. В. – д-р техн. наук, профессор, профессор кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники, Харьков, Украина.

Плисс И. П. – канд. техн. наук, ведущий научный сотрудник ПНДЛ АСУ Харьковского национального университета радиоэлектроники, Харьков, Украина.

Шафроненко А. Ю. – канд. техн. наук, доцент, доцент кафедры информатики Харьковского национального университета радиоэлектроники, Харьков, Украина.

АННОТАЦИЯ

Актуальность Задача кластеризации массивов наблюдений произвольной природы является неотъемлемой частью Data Mining, а в более общем случае Data Science, для ее решения предложено огромное количество подходов, отличающихся между собой как априорными предположениями относительно физической природы данных и задачи, так и математическим аппаратом. С вычислительной точки зрения задача кластеризации превращается в проблему поиска локальных экстремумов многоэкстремальной функции векторного аргумента плотности с помощью градиентных процедур, многократно запускаемых с разных точек исходного массива данных. Ускорить процесс поиска этих экстремумов можно, воспользовавшись идеями эволюционной оптимизации, включающей в себя алгоритмы, инспирированные природой, роевые алгоритмы, популяционные алгоритмы и т.д.

Цель. Цель работы заключается во внедрении процедуры кластеризации данных на основе пиков плотности распределения данных и эволюционного метода кошачьих стай, объединяющей в себе основные преимущества методов работы с данными в условиях пересекающихся классов, характеризуется качественной кластеризацией, высоким быстродействием и точностью полученных результатов.

Метод. Введен метод кластеризации массивов данных на основе комбинированной оптимизации функций плотности распределения и эволюционного метода кошачьих стай. Преимуществом предлагаемого подхода является сокращение времени решения оптимизационных задач в условиях, когда кластеры пересекаются.

Результаты. Результаты экспериментов подтверждают эффективность предлагаемого подхода в задачах кластеризации при условии классов, которые пересекаются и позволяют рекомендовать предложенный метод для использования на практике для решения проблем автоматической кластеризации больших данных.

Выводы. Введен метод кластеризации массивов данных на основе комбинированной оптимизации функций плотности распределения и эволюционного метода кошачьих стай. Преимуществом предлагаемого подхода является сокращение времени решения оптимизационных задач в условиях, когда кластеры пересекаются. Метод достаточно прост с точки зрения численной реализации и не является критическим для выбора оптимизационной процедуры. Результаты экспериментов подтверждают эффективность предлагаемого подхода в задачах кластеризации в условиях пересекающихся кластеров.

КЛЮЧЕВЫЕ СЛОВА: нечеткая кластеризация, пики плотности распределения данных, эволюционный метод.

UDC 004.8:004.032.26

CLUSTERIZATION OF DATA ARRAYS BASED ON COMBINED OPTIMIZATION OF DISTRIBUTION DENSITY FUNCTIONS AND THE EVOLUTIONARY METHOD OF CAT SWARM

Bodyanskiy Ye. V. – Dr. Sc., Professor at the Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Pliss I. P. – PhD, Leading Researcher at Control Systems Research Laboratory, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

DOI 10.15588/1607-3274-2022-4-5

Shafronenko A. Yu. – PhD, Associate Professor at the Department of Informatics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. The task of clustering arrays of observations of an arbitrary nature is an integral part of Data Mining, and in the more general case of Data Science, a huge number of approaches have been proposed for its solution, which differ from each other both in a priori assumptions regarding the physical nature of the data and the problem, and in the mathematical apparatus. From a computational point of view, the clustering problem turns into a problem of finding local extrema of a multiextremal function of the vector density argument using gradient procedures that are repeatedly launched from different points of the initial data array. It is possible to speed up the process of searching for these extremes by using the ideas of evolutionary optimization, which includes algorithms inspired by nature, swarm algorithms, population algorithms, etc.

Objective. The purpose of the work is to introduce a data clustering procedure based on the peaks of the data distribution density and the evolutionary method of cat swarms, that combines the main advantages of methods for working with data in conditions of overlapping classes, is characterized by high-quality clustering, high speed and accuracy of the obtained results.

Method The method for clustering data arrays based on the combined optimization of distribution density functions and the evolutionary method of cat swarms was proposed. The advantage of the proposed approach is to reduce the time for solving optimization problems in conditions where clusters are overlap.

Results. The results of the experiments confirm the effectiveness of the proposed approach in clustering problems under the condition of classes that overlap and allow us to recommend the proposed method for use in practice to solve problems of automatic clustering big data.

Conclusions. The method for clustering data arrays based on the combined optimization of distribution density functions and the evolutionary method of cat swarm was proposed. The advantage of the proposed approach is to reduce the time for solving optimization problems in conditions where clusters are overlap. The method is quite simple from the numerical implementation and is not critical for choosing an optimization procedure. The experimental results confirm the effectiveness of the proposed approach in clustering problems under conditions of overlapping clusters.

KEYWORDS: fuzzy clustering, density peak of dataset, evolutionary method.

REFERENCES

1. Gan G. Ma Ch., Wu J..Data Clustering: Theory, Algorithms and Applications. Philadiphia, Pensilvania, SIAM, 2007, 455 p. DOI: https://doi.org/10.1137/ 1.9780898718348
2. Abonyi J., Feil D. Cluster Analisis for Data Mining and System Identification. Basel, Birlhause, 2007, 303 p.
3. Xu R., Wunsch D. C. Clustering. Hoboken N. J., John Wiley & Sons, Inc., 2009, 398 p.
4. Aggarwal C. C. Data Mining. Switzerland, Springer, 2015, 727 p. DOI https://doi.org/ 10.1007 / 978-3-319-14142-8.
5. Engelbrecht A. Computational intelligence: an introduction. Sidney, John Wiley & Sons, 2007, 597 p.
6. Rutkowski L. Computational Intelligence Methods and Techniques. Berlin Heidelberg, Springer-Verlag, 2008, 514 p.
7. Kroll A. Computational Intelligence. Eine Einfürung in Problelme, Methoden and Tchnische Anwendungen. München, Oldenbourg Verlag, 2013, 428 p.
8. Kohonen T. Self-Organizing Maps. Berlin, Springer, 1995, 362 p. DOI: 10.1007/978-3-642-56927-2.
9. Hinneburg A., Klein D. An efficient approach to clustering in large multimedia databases with noise, Proc. 4th Int. Conf. in Knowledge Discovering and Data Mining, KDD98, N.Y., AAAI Press, Aug. 27, 1998. Hinneburg, 1998, pp. 58–65.
10. Hinneburg A., Gabriel H. H. Denclue 2.0: Fast clustering based on kernel density estimation, *International symposium on intelligent data analysis*. Springer, Berlin, Heidelberg, 2007, pp. 70–80. https://doi.org/10.1007 /978-3-540-74825-0_7
11. Hinneburg A., Keim D. A. A general approach to clustering in large databases with noise-knowledge and Identification Systems, 2003, 5 (4), pp. 387–415. https://doi.org/10.1007 /s10115-003-0086-9
12. Rehioui H. et al. DENCLUE-IM: A new approach for big data clustering, *Procedia Computer Science,* 2016, Vol. 83, pp. 560–567. DOI: 10.1016/j.procs.2016.04.2 65
13. Epanechnikov V. A. Nonparametric estimation of multivariate probability density, *Probability theory and its Application*, 1968, 14, No. 2, pp. 156–161.
14. Nadaraya E. A. On non-parametric estimates of density functions and regression curves, *Theory of Probability & Its Applications*, 1965, Vol. 10, No. 1, pp. 186–190.
15. Watson G. S. Smooth regression analysis, Sankhyā: The Indian Journal of Statistics, Series A, 1964, pp. 359–372.
16. Grosan C., Abraham A., Chis M. Swarm intelligence in Data Mining, *Studies in Computational Intelligence,* 2006, № 34, pp. 1–20.
17. Shafronenko A. Yu, Bodyanskiy Ye. V., Pliss I. P. The Fast Modification of Evolutionary Bioinspired Cat Swarm Optimization Method [Electronic resource], 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), 2019. Sozopol, Bulgaria, 2019, pp. 548–552. DOI: 10.1109 /CAOL46282. 2019.9019583
18. Eiben A., Smith J. Introduction to Evolutionary Computing. Heidelberg, Springer, 2003.
19. Karpenko A. P. Population algorithms for global continious optimization. Review of new and little-known algorithms, *Supplement to the journal "Information, Technologies",* 2012, No. 7, 32 p.
20. Chu S.-C., Tsai P.-W., Pan J. S. Cat swarm optimization, *Lecture Notes in Artificial Intelligence,* 4099. Berlin Heidelberg, Springer-Verlag, 2006, pp. 854–858.
21. Chu S.-C., Tsai P.-W. Computational Intelligence based on the behavior of cats, *International Journal of Innovative Computing, Information, and Control,* 2007, Vol. 3, № 1, pp. 163–173.
22. Shafronenko A., Bodyanskiy Ye., Pliss I., Klymova I. Online Credibilistic Fuzzy Clustering Method Based on Cauchy Density Distribution Function, *2021 11th International Conference on Advanced Computer Information Technologies (ACIT), proceedings.* Deggendorf, Germany, IEEE, 2021, pp. 704–707. DOI: 10.1109/ ACIT52158.2021.9548572

DOI 10.15588/1607-3274-2022-4-5